

**WHAT IS CLAIMED IS:**

1. A context switching system for a multi-thread execution  
2 pipeline loop having a pipeline latency, comprising:

3 a context switch requesting subsystem configured to:

4 detect a device request from a thread executing within  
5 said multi-thread execution pipeline loop for access to a  
6 device having a fulfillment latency exceeding said pipeline  
latency, and

7 generate a context switch request for said thread; and

8 a context controller subsystem configured to receive said  
9 context switch request and prevent said thread from executing until  
10 said device request is fulfilled.

11 2. The context switching system as recited in Claim 1  
wherein said context controller subsystem is further configured to  
allow said thread to continue to traverse said multi-thread  
execution pipeline loop while waiting for said device request to be  
fulfilled.

3. The context switching system as recited in Claim 1  
wherein said context controller subsystem is further configured to  
allow other threads within said multi-thread execution pipeline  
loop to continue to execute while said thread is waiting for said  
device request to be fulfilled.

4. The context switching system as recited in Claim 1  
2 further comprises a miss fulfillment first-in-first-out buffer  
3 (FIFO), said context controller subsystem further configured to  
4 employ said FIFO to:

5 store said thread in said miss fulfillment FIFO upon reaching  
6 an end position of said multi-thread execution pipeline loop,  
7 sequence said thread through said miss fulfillment FIFO, and  
reinsert said thread into said multi-thread execution pipeline  
loop at a beginning position.

5. The context switching system as recited in Claim 4  
wherein said context controller subsystem is further configured to  
store said thread in said miss fulfillment FIFO upon receiving said  
context switch request.

6. The context switching system as recited in Claim 1  
2 wherein said context controller subsystem is further configured to  
3 replace said thread's current instruction with a NO-Operation (NOP)  
4 instruction to prevent said thread from executing until said device  
5 request is fulfilled.

7. The context switching system as recited in Claim 1  
2 wherein said device request is a request to access external memory  
3 due to a cache miss status.

8. For use with a multi-thread execution pipeline loop  
2 having a pipeline latency, a method of operating a context  
3 switching system, comprising:

4 detecting a device request from a thread executing within said  
5 multi-thread execution pipeline loop for access to a device having  
6 a fulfillment latency exceeding said pipeline latency;

7 generating a context switch request for said thread when said  
8 thread issues said device request; and

9 receiving said context switch request and preventing said  
10 thread from executing until said device request is fulfilled.

9. The method as recited in Claim 8 further comprising  
allowing said thread to continue to traverse said multi-thread  
execution pipeline loop while waiting for said device request to be  
fulfilled.

10. The method as recited in Claim 8 further comprising  
2 allowing other threads within said multi-thread execution pipeline  
3 loop to continue to execute while said thread is waiting for said  
4 device request to be fulfilled.

11. The method as recited in Claim 8 further comprising  
2 employing a miss fulfillment first-in-first-out buffer (FIFO) for:  
3 storing said thread in said miss fulfillment FIFO upon  
4 reaching an end position of said multi-thread execution pipeline  
5 loop,

6 sequencing said thread through said miss fulfillment FIFO, and  
7 reinserting said thread into said multi-thread execution  
pipeline loop at a beginning position.

12. The method as recited in Claim 11 wherein said storing  
further comprises storing said thread in said miss fulfillment FIFO  
upon receiving said context switch request.

13. The method as recited in Claim 8 wherein said preventing  
2 further comprises replacing said thread's current instruction with  
3 a NO-Operation (NOP) instruction to prevent said thread from  
4 executing until said device request is fulfilled.

14. The method as recited in Claim 8 wherein said device  
2 request is a request to access external memory due to a cache miss  
3 status.

15. A fast pattern processor that receives and processes

2 protocol data units (PDUs), comprising:

3 a dynamic random access memory (DRAM) that contains  
4 instructions;

5 a memory cache that caches certain of said instructions from  
6 said DRAM; and

7 a tree engine that parses data within said PDUs and employs  
8 said DRAM and said memory cache to obtain ones of said  
instructions, including:

10 a multi-thread execution pipeline loop having a pipeline  
11 latency, and

12 a context switching system for said multi-thread  
13 execution pipeline loop, having:

14 a context switch requesting subsystem that:

15 detects a device request from a thread  
16 executing within said multi-thread execution  
17 pipeline loop for access to a device having a  
18 fulfillment latency exceeding said pipeline  
19 latency, and

20 generates a context switch request for  
21 said thread, and

22 a context controller subsystem that receives said  
23 context switch request and prevents said thread from  
24 executing until said device request is fulfilled.

16. The fast pattern processor as recited in Claim 15 wherein  
2 said context controller subsystem further allows said thread to  
3 continue to traverse said multi-thread execution pipeline loop  
4 while waiting for said device request to be fulfilled.

17. The fast pattern processor as recited in Claim 15 wherein  
said context controller subsystem further allows other threads  
within said multi-thread execution pipeline loop to continue to  
execute while said thread is waiting for said device request to be  
fulfilled.

18. The fast pattern processor as recited in Claim 15 wherein  
2 said context switching system further includes a miss fulfillment  
3 first-in-first-out buffer (FIFO), said context controller subsystem  
4 employs said FIFO to:

5 store said thread in said miss fulfillment FIFO upon reaching  
6 an end position of said multi-thread execution pipeline loop,  
7 sequence said thread through said miss fulfillment FIFO, and  
8 reinsert said thread into said multi-thread execution pipeline  
9 loop at a beginning position.

19. The fast pattern processor as recited in Claim 18 wherein  
2 said context controller subsystem stores said thread in said miss  
3 fulfillment FIFO upon receiving said context switch request.

20. The fast pattern processor as recited in Claim 15 wherein  
2 said context controller subsystem replaces said thread's current  
3 instruction with a NO-Operation (NOP) instruction to prevent said  
thread from executing until said device request is fulfilled.

21. The fast pattern processor as recited in Claim 15 wherein  
said device is said DRAM and said device request is a request to  
access said DRAM due to a cache miss status from said memory cache.